VFAST Transactions on Software Engineering
http://vfast.org/journals/index.php/VTSE@ 2018, ISSN (e): 2309-6519; ISSN (p): 2411-6327
Volume 13, Number 2, May-August, 2018 *pp:60-67*

A COMPARATIVE ANALYSIS OF DISTRIBUTED AND PARALLEL COMPUTING

KHAWAJA UBAID UR REHMAN[1], KH. RAEES UR REHMAN[2], ADEEL ASHRAF[1]

[1]Department of Computer Science, University of Management and Technology, Lahore, Punjab, Pakistan
[2]Services Institute of Medical Sciences, Lahore, Punjab, Pakistan
Email: ubaid.rehman@umt.edu.pk

ABSTRACT. *In the age of emerging technologies, the amount of data is increasing very rapidly. Due to massive increase of data the level of computations are increasing. Computer executes instructions sequentially. But the time has now changed and innovation has been advanced. We are currently managing gigantic data centers that perform billions of executions on consistent schedule. Truth be-hold, if we dive deep into the processor engineering and mechanism, even a successive machine works parallel. Parallel computing is growing faster as a substitute of distributing computing. The performance to functionality ratio of parallel systems is high. Also, the I/O usage of parallel systems is lower because of ability to perform all operations simultaneously. On the other hand, the performance to functionality ratio of distributed systems is low. The I/O usage of distributed systems is higher because of incapability to perform all operations simultaneously. In this paper, an overview of distributed and parallel computing is described. The basic concept of these two computing is discussed. In addition to this, pros and cons of distributed and parallel computing models are described. Through many aspects, we can conclude that parallel systems are better than distributed systems.*
Keywords*:* Distributed computing; Parallel computing

1. **Introduction.** The computing field is one of the rapidly emerging industries and it is induced by the fastest technological advancements in the regions of computer hardware and software. The technological developments in hardware particularly in chip manufacturing and fabrication advancements, rapid and cheap microprocessors in addition to this there is high signal and low lag interconnection networks advancement. Among these the latest advancements in electronics has played a pivotal role in development of great sequential and parallel computers.

Software technology is also progressing swiftly. Today there is availability of mature software's. This has empowered the advancement of applications dealing in scientific and commercial necessities. The need of powerful parallel computers for weather forecasting and earth quake prediction has become necessity. The term distributed computing initially referred to computer networks where single computer was actually located within some geographical area [1]. Now the term distributed computing is used in broader sense, it is a branch of computer science which deals with distributed systems. Distributed computing involves multiple computers distant from each other. Distributed computers perform role in computation and information processing problems. It uses a number of network computers for accomplishing a task. To achieve computational result more faster as compare to single computer concept of distributed computing was introduced. The networked computers can be in a same apartment, same site, and same state or in different counties. Distributed systems are being thought of as conventional networks of independent computers. A distributed system can be a combination of MPP and SMP, clusters and individual computers.

In parallel computing many processors perform or process an application or computation concurrently [2]. Parallel computing helps in accomplishing vast calculations by distributing the major problems into minor ones which can be resolved at an equal period. Parallel computing consumes two or more than two processors (core, computers) in addition to resolve a single question. There are numerous forms of parallel computing, but the awareness has matured quite lately because of physical restrictions. Power utilization (and thus heat production) using computers has developed an apprehension in current years, parallel computing has converted the significant model in computer architecture, primarily in the mold of multi-core processors [3].

In this paper, an overview of distributed and parallel computing is given. The basic concept of these two computing is discussed. In addition to this, pros and cons of distributed and parallel computing models are discussed. The view of distributed vs. parallel computing is shown below in Figure-1.

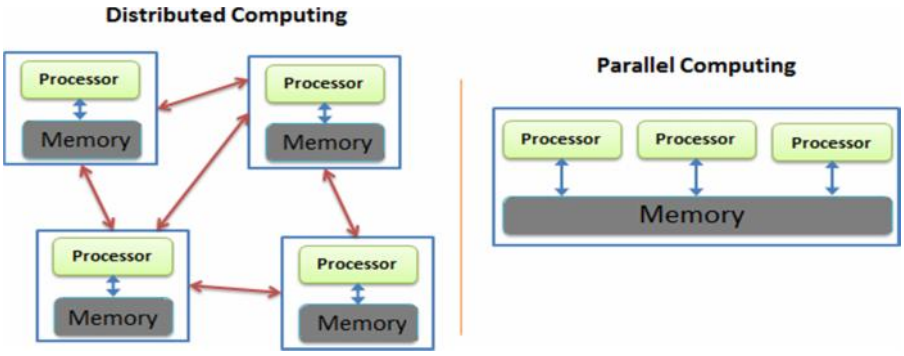

Figure 1:- Distributed vs. Parallel Computing

2. Related Work. The first widely used distributed systems were LAN i.e. Ethernet that was created in the mid-1970s [4]. A forerunner of Internet i.e. ARPANET was presented in the former 1960s. In the early 1970's ARPANET email was invented. Enormous parallel architecture began rising and communication passing interface was established. Bandwidth was a major concern. The application of ARPANET i.e. email became more successful [5]. This application was the first huge scale distributed application. Besides its descendant Internet initial worldwide computer networks comprised of Usenet and Fido Net which both supports distributed systems. The former most distributed computing project was based on Internet. It began after 1987 by the famous most DEC System Research Center. Distributed.net was a scheme launched in 1977 [5]. It was of its unique kind to use the Internet to disseminate information for calculating and gathering outcomes. From 1955 till today Cluster/Grid architecture has massively increased.

Michael J. Flynn's designed an initial taxonomy to distinguish multi-processor computer architectures to classify two non-dependent dimensions i.e. instruction & data. Two of these dimensions can have single or two states i.e. One or Multiple. The probable categorizations given by Flynn's taxonomy are: SISD, SIMD, MIMD and MIMD [5].

From 1986 speed of the computer architecture was doubled by the arrival of very large scale integration (VLSI). Incrementing the size of word decreases the sum of instructions. Greater the word size shorter the number of instructions and vice versa [6]. Generally, four bit microchips were exchanged with eight bit, there after sixteen bit with thirty two bit microchips. In recent most times, with development of 32 and 64 bit architectures, 64-bit CPUs became more common in use.

A computer is a set of commands performed by the processor. These commands can be re-arranged and joined into clusters which are then run in parallel without changing the outcome of the program. Present processors have multiple pipelines of instructions. The PC having n-phase pipeline will have up to n varied number of commands at altered phases of consummation [6].

Parallel programs has a feature of task parallelism in which solely unalike calculations can be performed either on the similar/dissimilar collections of data. Whereas in data parallelism identical calculations are executed on the similar/dissimilar collections of data [7].

In parallel computer the main memory is shared but in distributed computer the main memory is cogently distributed [8]. Such systems in which every component of the main memory is accessed with equal latency and bandwidth is UMA and the systems in which memory access time hinge on the memory associated to the processor is known as NUMA architecture. Distributed systems have NUMA architectures for memory.

Parallel systems have problems with caches that pile the same value at more than one place. Such systems need a cache coherency which saves the track of cached values. Bus snooping is a technique which keeps the record of the cached values. In hardware, communication between processor & memory, processor to processor can be implemented in various ways like shared bus and cross bar switch etc. [8]. Parallel computers enable passing of messages between nodes which are not linked directly. The method used for communication between PCs is hierarchical.

3. Technical Framework.

1. Distributed System Architectures

There are various architectures used for distributed computing. Distributed computing programming models falls into several types which are explained below:

a. Client-server architecture

In client-server architectural model client receives services whereas server provides services. The client needs a service and server gives the services. The client service model divides the tasks or loads between the service requester and provider i.e. client and server respectively as shown in Figure-2. Normally client and server communicate via a network through a distinct hardware but individually client and server can be present in identical systems [13]. Client server has many advantages few of them are easy maintenance, integration of services, resource sharing among different platforms and improved data sharing. The disadvantages of client server architecture are server gets severely overloaded on regular concurrent requests. Also client server model is centralized if a server fails, the client requests will get failed and the task will not get accomplished.

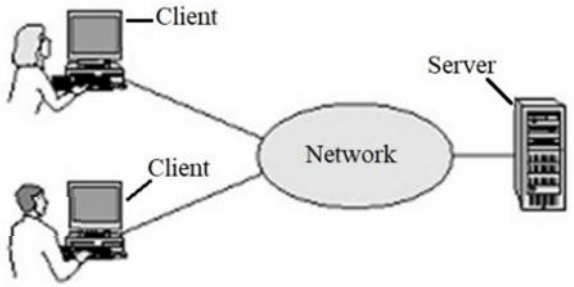

Figure 2:- Client Server architecture

b. Three-tier architecture

Three-tier architecture includes client, business and data tier [10]. Layers of three tier architecture are explained and shown in Figure-3.

i. Client layer

This layer comprises of UI of an application. It is used for the front end purpose. This layer contains the design part where the data and UI is shown to the user.

ii. Business layer

The business layer contains the business logics. It is a layer where business rules are created.

iii. Data Layer

The data layer contains how to insert, retrieve, update and delete data from database. It involves the functions to connect to the database and perform CRUD cases.

Three-tier architecture has many advantages few of them are it can be upgraded or replaced independently and business logic is more secure because client does not have direct access to the database. Three-tier architecture has many disadvantages few of them are the physical partitioning between the tiers may affect the performance and it is very difficult to set up & maintain.

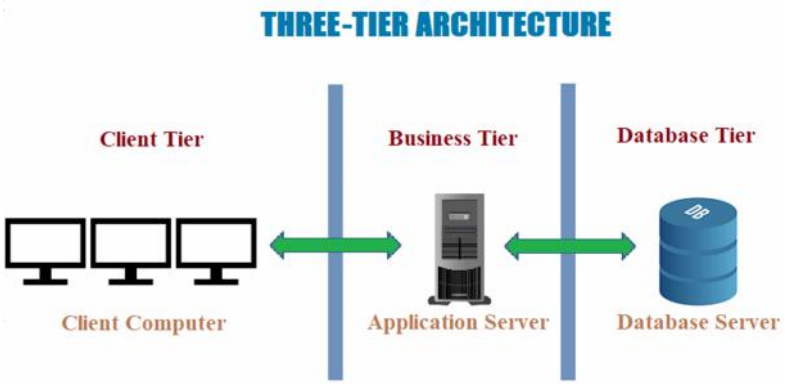

Figure 3:- Three tier architecture

c. N-tier architecture

N tier architecture is a multi-tier architecture in which data management and presentation functions are physically & logically alienated. It is also divided into presentation layer, business layer and data access layer and data layer [11], [12] as shown in Figure-4. N-tier architecture is identical to three-tier architecture in which middle layer and data layer is divided into new layer. This kind of application is responsible for the triumph of application servers.

An N-tier architecture has many advantages few of them are: N-tier applications are easier to maintain, N-tier

applications are highly interoperable, using an N-tier architecture developer is free to develop components of different layers with flexibility and new features can be added to the existing application without affecting the performance. The disadvantage of N tier architecture is that it requires lots of design to ensure integrity between nodes.

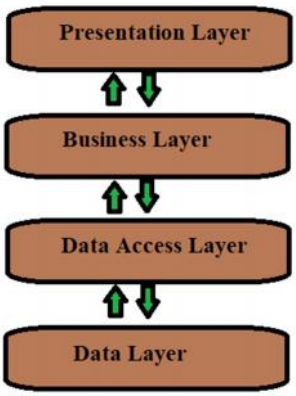

Figure 4:- N-tier architecture

d. Tightly coupled architecture

Tightly coupled architecture is a collection of integrated machines in which the same operation is performed in parallel. It is done by dividing the task in parts individually and they are put back for the final computation of the result [13].

e. P2P architecture

P2P stands for Peer to Peer. In this architecture different machines provides services or manages the network resources. All the tasks are equally distributed amid all machines known as peers. Peers can be client as well as servers [13]. The P2P architecture nodes are connected to each other to share resources without any central controlling server as shown in Figure-5. A P2P architecture has many advantages few of them are: P2P architecture is simple, P2P systems are inexpensive and P2P systems are more reliable because there is no central dependency. If one peer fails it does not affect others. The disadvantages of P2P architecture are: P2P architecture is not secure. Viruses, Trojans etc. can easily be spread in P2P architectures and for P2P architecture each user should be trained to perform administrative tasks.

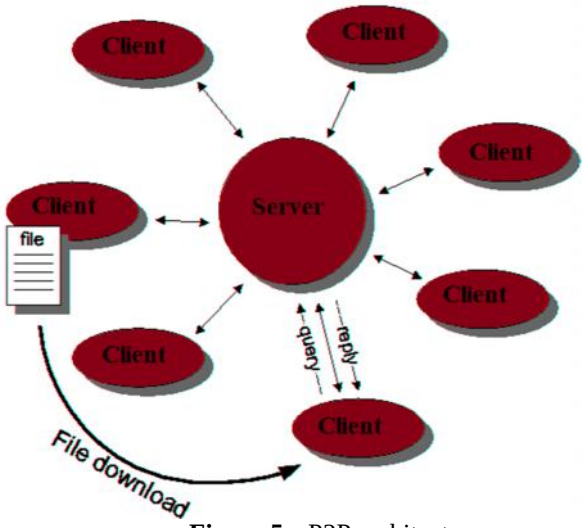

Figure 5:- P2P architecture

f. Parallel System Architectures

Parallel computing helps in distributing the bigger problems into smaller problems which can be solved simultaneously. Parallel computers uses two or more processors (core, computers) to solve a single problem. Most widely used taxonomy for parallel computer systems is Flynn's taxonomy. To distinguish between multi-processor computers architectures he classified into two non-dependent dimensions i.e. instruction & data. The categorizations given by Flynn's taxonomy are: SISD, SIMD, MIMD and MIMD [9] which are explained below.

g. SISD architecture

A non-parallel computer. SISD stands for Single Instruction Single Data Stream. In SISD a single instruction is executed on a single data stream. A single instruction stream is executed by CPU in a single clock cycle and single data stream is used as an input at one clock cycle. There are many advantages of the SISD architecture few of them are: It

provides faster execution for sequential instructions and extra resources are not required [14]. The advantages of SISD architecture is that large operations are executed slowly using SISD [14].

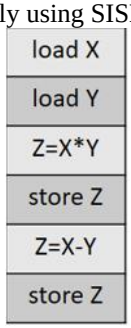

Figure 6:- Operations on SISD Machine

2) SIMD architecture

A parallel computer. SIMD stands for Single Instruction Multiple Data. In SIMD single instruction is executed on multiple data streams. In SIMD same instruction is performed at a certain clock cycle. Each processor run on dissimilar data elements. Two varieties of SIMD are processor arrays and vector pipelines. This architecture is best suited for graphics/image processing. There are many advantages of the SIMD architecture few of them are: High precision and scalability of size & performance [14]. The disadvantage of SIMD architecture is lack of applicability to a wide variety of problems [14].

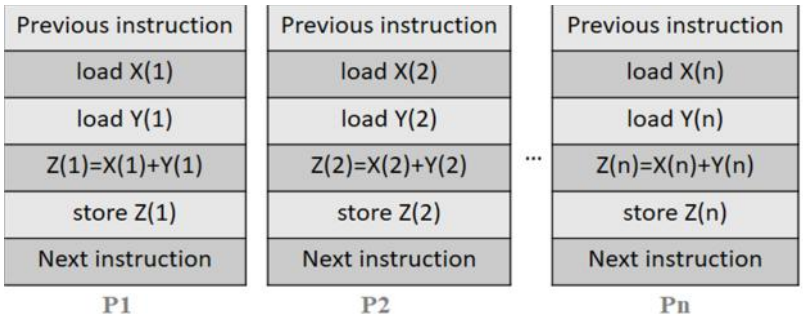

Figure 7:- Operations on SIMD Machine

3)MISD architecture

A type of parallel computer. MISD stands for Multiple Instruction Single Data. Multiple instructions are executed on single data stream in MISD. In MISD there are different instructions streams which operates on single data as shown in Figure-8. A single data is input to different processing elements. The advantage of MISD architecture is that it can be used in high end parallel environment [14]. The disadvantage is that many processors might remain idle in fewer instructions [14].

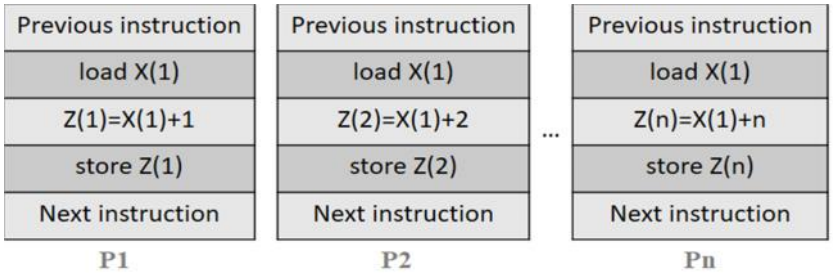

Figure 8:- Operations on MISD Machine

4)MIMD architecture

A type of parallel computer. MIMD stands for Multiple Instruction Multiple Data. In MIMD multiple instructions are executed on multiple data streams. In MIMD each processor may execute a different instruction stream. Every processor work with various data streams as shown in Fig 9. Execution can be synchronous or asynchronous, deterministic or non-deterministic. The advantages of MIMD are that it is reliable, scalable and efficient [14]. The disadvantage of MIMD architecture is that it is costly and sometimes complex to implement [14].

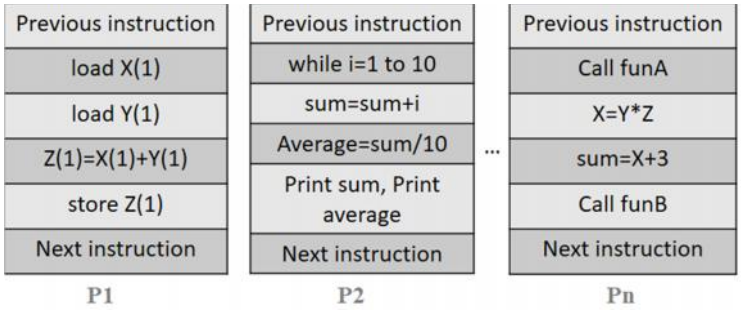

Figure 9:- Operations on MIMD Machine

4. Comparative Analysis. The benefit of distributed systems over parallel systems is in-built support of fault tolerance. Distributed systems are designed to support fault tolerance as one of the core objectives whereas parallel systems do not provide in-built support of fault tolerance [15]. Numerous work has been done on adding fault tolerance support to parallel systems however there is no such achievement.

Parallel systems supports high communication protocols like Gemini, Infini-band other than Ethernet. Due to this reason parallel systems provides high end performance. Another important characteristics of distributed systems is level of abstraction. It provides high level of abstraction to users. Users can image data structures for instance array list as distributed array list. Parallel systems provide lower level of abstraction but there are some frameworks like Charm++ and HPX which provides high level abstraction [15].

The memory usage of distributed systems is higher due to data immutability it means for each operation it creates new copy of objects which consumes more memory. For some applications memory usage can be problematic as it creates new copies of data which can be a limiting factor whereas parallel systems has low memory usage. Nodes in distributed systems serve and respond to many interactive users. Classically there are one or more users per node in distributed systems whereas in parallel systems there are several nodes dedicated to a single computer [15].

In distributed systems applications exploits parallelism at program level where programs are distributed between set of available nodes. Parallel systems exploits parallelism in which application uses a huge set of nodes and communicates often as compared to distributed programs. Applications of parallel systems are self-synchronizing. The data exchange between nodes happen at equal intervals. If the processes do not run simultaneously resources will get wasted. Parallel systems supports gang scheduling whereas distributed systems are not self-synchronizing [15].

I/O usage of distributed systems is higher because tasks are distributed at each iteration. I/O usage of parallel systems is low because tasks are performed simultaneously. The time to develop an application using distributed frameworks like Spark and Flink require skills of lower level because API's and programming interface of distributed frameworks are simple. If the main target is to get solution rapidly and at lower cost than frameworks like Spark and Flink are favorable. On the contrary to this it is not the case with parallel systems [15].

Another important area of concern for parallel and distributed systems is dynamic resource utilization. The systems should be able to scale up and down dynamically upon the workload and amount of resources (number of nodes). Distributed systems are less homogeneous than parallel systems. This is because if we talk about a workstation then it has different CPU's, performance rating, architectures, operating systems and configurations [15].

From all above discussion the following analysis between Distributed and Parallel systems are concluded which is shown in Table 1.

Table 1. Distributed vs. Parallel Systems

Sr. #	Distributed systems	Parallel systems
1	Distributed systems are fault tolerant for instance Distributed systems framework RDD sparks provides fault tolerance.	Parallel systems do not support fault tolerance.
2	Distributed systems usually supports protocols like Ethernet.	Parallel systems have various protocols for instance Gemini and Ethernet etc.
3	Distributed systems has higher level of abstraction.	Parallel systems supports both high level abstraction and low level abstraction but commonly they have low level abstraction.
4	Memory usage of distributed systems is higher due to data	Memory usage of parallel systems are lower due to few overheads and

	immutability.	optimized functions.
5	The performance to functionality ratio of distributed systems is low.	The performance to functionality ratio of parallel systems is high.
6	The user per node ratio of distributed systems is high	The user per node ratio of parallel systems is less.
7	Parallelism in distributed systems is coarse grained	Parallelism in parallel systems is fine grained
8	There is no gang scheduling in distributed systems	There is gang scheduling in parallel systems
9	I/O usage of distributed systems is higher because of incapability to perform all operations simultaneously.	I/O usage of parallel systems is lower because of ability to perform all operations simultaneously.
10	The time to develop a code is comparatively lower because of simple API's.	The time to develop a code in parallel systems is higher because it is more complicated.
11	The level of skills required to develop an application is lower due to simple API's.	The level of skills required to develop an application is higher because it requires good understanding of parallel computing concepts.
12	Dynamic resources are available in many distributed systems frameworks.	Dynamic resources are available in some parallel systems frameworks.
13	In distributed systems memory is loosely coupled	In parallel systems memory is tightly coupled.
14	Distributed systems are less homogeneous because of different physical memory, possibly different systems and varying configuration.	Parallel systems are homogeneous. They have different amounts of memory otherwise it is similar.
15	Distributed systems can be spread over a campus, or country through some physical media like Ethernet, radio links and satellite connections etc.	Parallel systems are contained in a single cabinet or collection of cabinets located in a single room.

Conclusion. In this paper, a detailed comparison on two computing models distributed vs. parallel is described. The problems associated with these computing models are described. Such a comparison in different aspects will help us to cognize the computing models since few features of these computing models are comparable. It also helps us in distinguishing the similarities and differences among these two computing. After all the above, it can be concluded that if we put more efforts on parallel computing the level of parallelism can be improved. We can also improve the computing speed by this. Also efforts are needed to put for the support of fault tolerance in parallel systems. Lot of efforts has been done in this specific field, yet at the same time more efforts are need to be done.

REFERENCES

[1] Lynch, N. A. (1996). Distributed algorithms. Morgan Kaufmann.

[2] Almasi, G. S., & Gottlieb, A. (1988). Highly parallel computing.

[3] Asanovic, K., Bodik, R., Catanzaro, B. C., Gebis, J. J., Husbands, P., Keutzer, K., ... & Yelick, K. A. (2006). The landscape of parallel computing research: A view from berkeley (Vol. 2). Technical Report UCB/EECS-2006-183, EECS Department, University of California, Berkeley.

[4] Andrews, G. R. (2000). Foundations of multithreaded, parallel, and distributed programming (Vol. 11). Reading: Addison-Wesley.

[5] El-Zoghdy, S. F., & Ghoniemy, S. (2014). A Survey of Load Balancing In High-Performance Distributed Computing Systems. International Journal of Advanced Computing Research, 1.

[6] Marlow, S. (2012). Parallel and concurrent programming in Haskell. In Central European Functional Programming School (pp. 339-401). Springer, Berlin, Heidelberg.

[7] Culler, D., Karp, R., Patterson, D., Sahay, A., Schauser, K. E., Santos, E., ... & Von Eicken, T. (1993, August). LogP: Towards a realistic model of parallel computation. In ACM Sigplan Notices (Vol. 28, No. 7, pp. 1-12). ACM.

[8] Hennessy, J. L., & Patterson, D. A. (2011). Computer architecture: a quantitative approach. Elsevier.

[9] Flynn, M. J., & Rudd, K. W. (1996). Parallel architectures. ACM Computing Surveys (CSUR), 28(1), 67-70.

[10] Eckerson, W. (1995). Three tier client/server architecture: Achieving scalability, performance and efficiency in client server applications. Open Information Systems, 10(1).

[11] Reig Ventura, G. (2008). Policy-Driven Resource Management for virtualized Grid providers.

[12] Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., & Stal, M. (1996). A system of patterns: Pattern-oriented software architecture.

[13] AL, M. K., TK, M. S., Kothiwale, M., & SS, K. Real Time And Distributed Computing Systems.

[14] Asghar, Z., & Ali, U. (2016). Comparative Analysis of Multiprocessor Architecture. International Journal of Advanced Research in Computer Science and Electronics Engineering (IJARCSEE), 5(12), pp-131.

[15] Riesen, R., Brightwell, R., & Maccabe, A. B. (1998). Differences between distributed and parallel systems. SAND98-2221, Unlimited Release, Printed October